

Simulasi Dan Visualisasi Algoritma Greedy Pemilihan Koin Dalam Bentuk Game

Eko Safitri UH¹ dan Wijanarto²

²Staff Pengajar Fakultas Ilmu Komputer
Universitas Dian Nuswantoro Semarang

Abstract

Money change is a common matter involving greedy algorithm principle in our daily lives. Greedy algorithm is one that solves problems step by step. Every step takes the best option without considering future consequence ("take what you can get now!" principle), expecting that by choosing local optimum in every step will result in global optimum. Simulation and visualization of education game of money change is the implementation of greedy algorithm in coin change. The implementation is in the form of web based simulation and simulation in the framework of algorithm design and analysis course.

Keyword: *algorithm, game, greedy, simulation, visualization*

1. Pendahuluan

Penukaran uang merupakan salah satu contoh masalah dalam kehidupan sehari-hari yang menggunakan prinsip algoritma *greedy*. Algoritma *greedy* adalah algoritma yang memecahkan masalah langkah per langkah, pada setiap langkah mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi kedepan (prinsip "*take what you can get now!*"), berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global. (1; 2)

Terdapat suatu masalah bagaimana menentukan algoritma untuk memberikan uang kembalian dari suatu pembayaran di super market dengan menggunakan sejumlah koin yang tersedia sekecil mungkin, urutan pemberian uang kembalian dimulai dari yang terbesar (3). Pada paper ini akan di implementasikan algoritma tersebut dalam bentuk visualisasi permainan (game visualitation) (4; 5; 6) berbasis web (7).

2. Kerangka Pemikiran

2.1. Algoritma Greedy

Algoritma *Greedy* adalah algoritma yang memecahkan masalah langkah per langkah, pada setiap langkah (2; 8; 9) akan melakukan satu, mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi kedepan (prinsip "*take what you can get now!*"), lalu berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global. Sementara elemen yang tersusun dari Algoritma *Greedy* (1; 10) adalah sebagai berikut,

- Himpunan Kandidat yang berisi elemen-elemen pembentuk solusi.
- Himpunan Solusi yang terdiri kandidat-kandidat yang terpilih sebagai solusi persoalan.
- Fungsi Seleksi (*selection function*) yaitu di pakai untuk memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada satu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
- Fungsi Kelayakan (*feasible*), berfungsi untuk memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (*constraints*) yang ada.

¹ Mahasiswa Teknik Informatika Fakultas Ilmu Komputer 2007

Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.

- e) Fungsi Obyektif merupakan fungsi yang memaksimalkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain).

2.2. Algoritma Greedy Pada Penukaran Uang

Pada masalah penukaran uang, elemen-elemen algoritma *greedy* –nya adalah (3; 2; 11) yang terdiri dari :

- Himpunan kandidat : himpunan koin yang merepresentasikan nilai 1,5,10,25, paling sedikit mengandung satu koin untuk setiap nilai.
- Himpunan solusi : total nilai koin yang dipilih tepat sama jumlahnya dengan nilai uang yang ditukarkan.
- Fungsi seleksi : pilihlah koin yang bernilai tertinggi dari himpunan kandidat yang tersisa.
- Fungsi layak : memeriksa apakah nilai total dari himpunan koin yang dipilih tidak melebihi jumlah uang yang harus dibayar.
- Fungsi obyektif : jumlah koin yang digunakan minimum.

Tabel 2.1 : *Pseudo-code* algoritma *greedy*

```

Procedure greedy (input C : himpunan_kandidat, OutputS :himpunan_solusi)
{menentukan solusi optimum dari persoalan optimasi dengan algoritma
greedy
Masukan : himpunan kandidat C
Keluaran : himpunan solusi S }
Deklarasi x : kandidat ;
Algoritma :
    S ← {}
    While (belum SOLUSI (S)) and (C ≠ {}) do
        x ← SELEKSI (C) ;
        C ← C - {x}
        If LAYAK (S U {x} ) then      S ← S U {x}
        Endif
    Endwhile
    {solusi (S) sudah diperoleh or C = {} }

```

Dalam menentukan solusi optimum dari persoalan optimasi dengan algoritma *greedy*. Pembuatan Prosedur Greedy dengan masukan berupa himpunan_kandidat dengan variable C, dan keluaran berupa himpunan_solusi dengan variable S. Variabel x dideklarasikan sebagai kandidat sementara yang dipilih.

Langkah algoritma *greedy* : inisialisasi awal himpunan_solusi dengan kosong. Selama himpunan solusi belum ditemukan dan himpunan-kandidat tidak kosong, maka akan dilakukan pemilihan sebuah kandidat dari himpunan_kandidat untuk mengisi variabel sementara x, sehingga himpunan_kandidat menjadi berkurang satu dan himpunan kandidat yang berkurang tadi menjadi himpunan kandidat yang baru. Jika anggota himpunan_solusi merupakan anggota himpunan x, maka himpunan solusi yang baru adalah anggota dari himpunan_solusi yang juga menjadi anggota dari himpunan x. Himpunan _solusi sudah diperoleh atau himpunan_kandidat =himpunan kosong (3; 1; 12).

2.3. Game

Game merupakan kata dalam bahasa inggris yang berarti permainan (13; 4; 14). Permainan itu sendiri adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang

menang dan ada yang kalah. Permainan ini biasanya dalam konteks tidak serius atau sebagai hiburan belaka. (6; 15)

2.4. Visualisasi

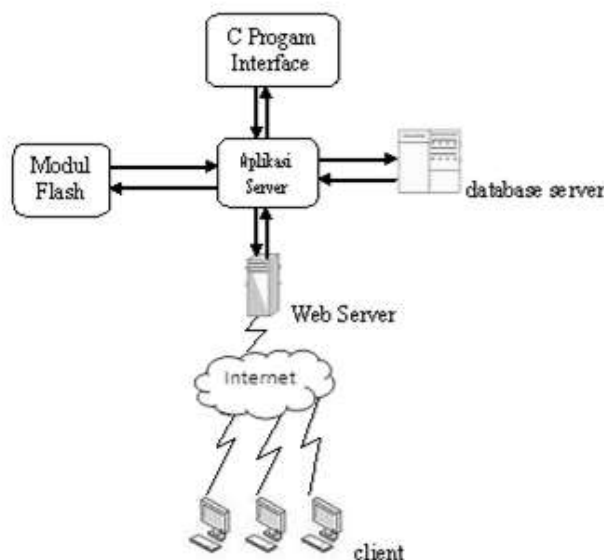
Visualisasi dalam bahasa Inggris “*visualization*” adalah rekayasa dalam pembuatan gambar, diagram, atau animasi untuk penampilan suatu informasi. Secara umum, visualisasi dalam bentuk gambar baik yang bersifat abstrak maupun nyata telah dikenal sejak awal dari peradaban manusia. Contoh dari hal ini meliputi lukisan dinding-dinding gua dari manusia purba, bentuk huruf hieroglif Mesir, sistem geometri Yunani, dan teknik pelukisan dari Leonardo da Vinci untuk tujuan rekayasa dan ilmiah. (6; 13)

2.5. Simulasi

Simulasi adalah suatu proses peniruan dari sesuatu yang nyata beserta keadaan sekelilingnya (*state of affair*). Aksi melakukan simulasi ini secara umum menggambarkan sifat-sifat karakteristik kunci dari kelakuan sistem fisik atau sistem yang abstrak tertentu (5). Model simulasi adalah metode penyampaian materi yang berbentuk permainan atau game yang meniru kondisi tertentu guna menanamkan materi pembahasan melalui pengalaman seseorang. Simulasi berguna untuk menambah atau meningkatkan keterampilan seseorang melalui “aktivitas tertentu” dalam kondisi tidak nyata.

3. Arsitektur aplikasi

Implementasi visualisasi dan simulasi algoritma greedy penukaran koin, terdiri dari desain arsitektur (13), arus data input dan output serta uji aplikasi. Di bawah ini dijelaskan arsitektur umum dari aplikasi yang di buat.



Gambar 3.1 : Arsitektur aplikasi

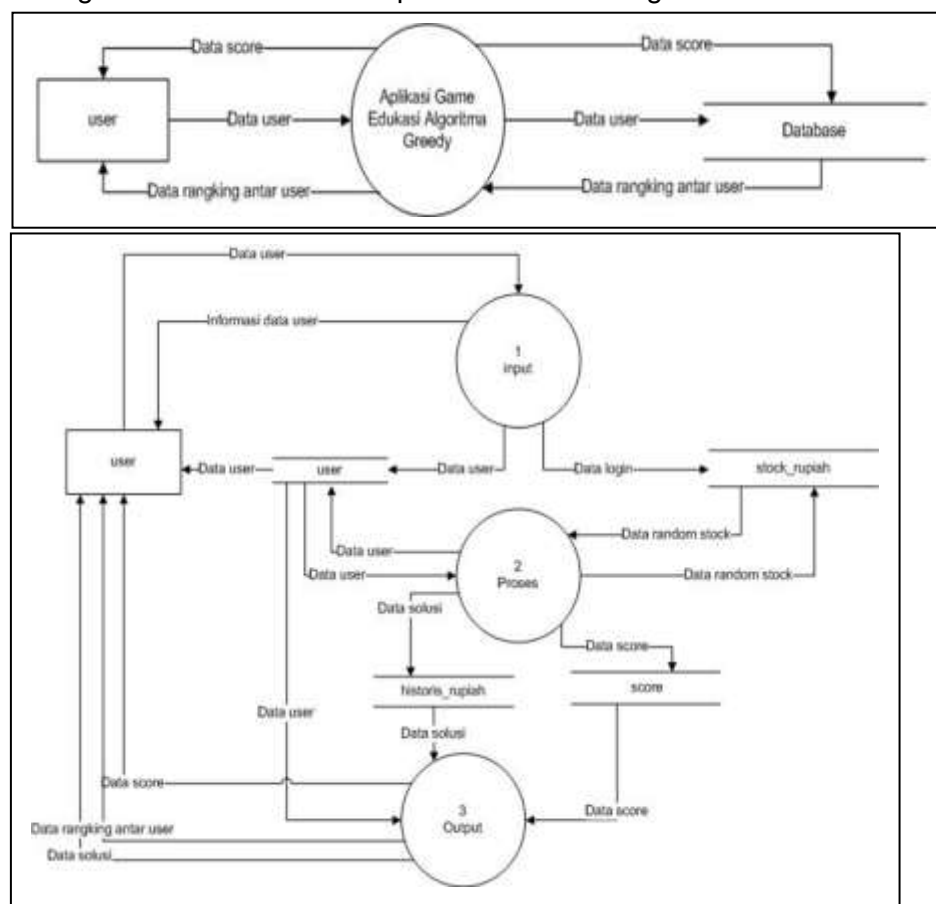
Dari Gambar di atas dapat dijelaskan sebagai berikut, Web server merupakan potongan perangkat lunak yang mendukung berbagai protokol Web, seperti HTTP untuk memproses permintaan client (5). Aplikasi server adalah aplikasi yang menangani fungsionalitas yang dibutuhkan oleh beberapa aplikasi, seperti workflow atau customization. Adobe Flash merupakan salah satu perangkat lunak komputer yang digunakan untuk membuat gambar vektor maupun animasi gambar

tersebut. Berkas yang dihasilkan mempunyai file extension .swf dan dapat diputar di penjelajah web yang telah dipasang Adobe Flash Player. Database server yaitu server yang biasanya menyuplai data produksi organisasi data dalam bentuk yang terstruktur, seperti didalam table. Client adalah browser (user agent) yang dikontrol oleh pengguna untuk mengoperasikan aplikasi web. Fungsionalitas client dapat dikembangkan dengan menginstal plug-in dan applet. C Interface Program (8) merupakan program dengan bahasa C yang dibuat untuk mencari 10 solusi optimum dalam pemilihan uang kembalian, sebagai jawaban optimal oleh komputer untuk di bandingkan dengan jawaban user.

Aplikasi Server / Aplikasi Web dari game edukasi algoritma greedy dibuat dengan PHP dan database MySQL. Tampilan utama Web adalah PHP, disini terdapat menu yang dihubungkan dengan modul-modul Flash seperti modul tutorial, asitektur dan permainan. Data-data dari modul flash permainan yang dibutuhkan untuk disimpan dalam database dikirim ke PHP kemudian dari PHP dikirim ke database. Sedangkan untuk mencari 10 solusi optimum menggunakan program C, data inputan untuk diolah di C diambil oleh PHP dari database. Dan output yang dihasilkan oleh C akan dibaca oleh PHP untuk ditampilkan di flash permainan. Aplikasi ini berada di Web server dan dapat diakses oleh client yang terhubung dengan internet.

3.1. Data Flow Diagram (7; 4; 5)

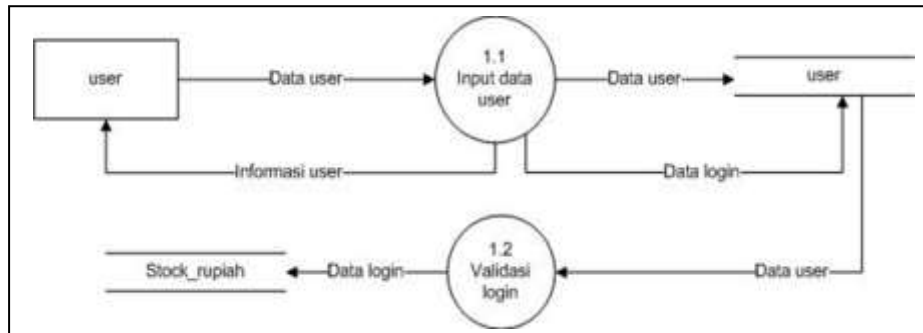
Secara ringkas aliran data dalam aplikasi di buat sebagai berikut :



Gambar 3.4 : DFD level 0 dan 1

Dari gambar di atas akan dijelaskan menjadi 3 bagian utama yaitu, Input, aplikasi disini akan menerima inputan dari user berupa data user yang dimana data user juga digunakan sebagai data login. Data user disimpan dalam tabel user dan data login disimpan dalam tabel stock_rupiah.

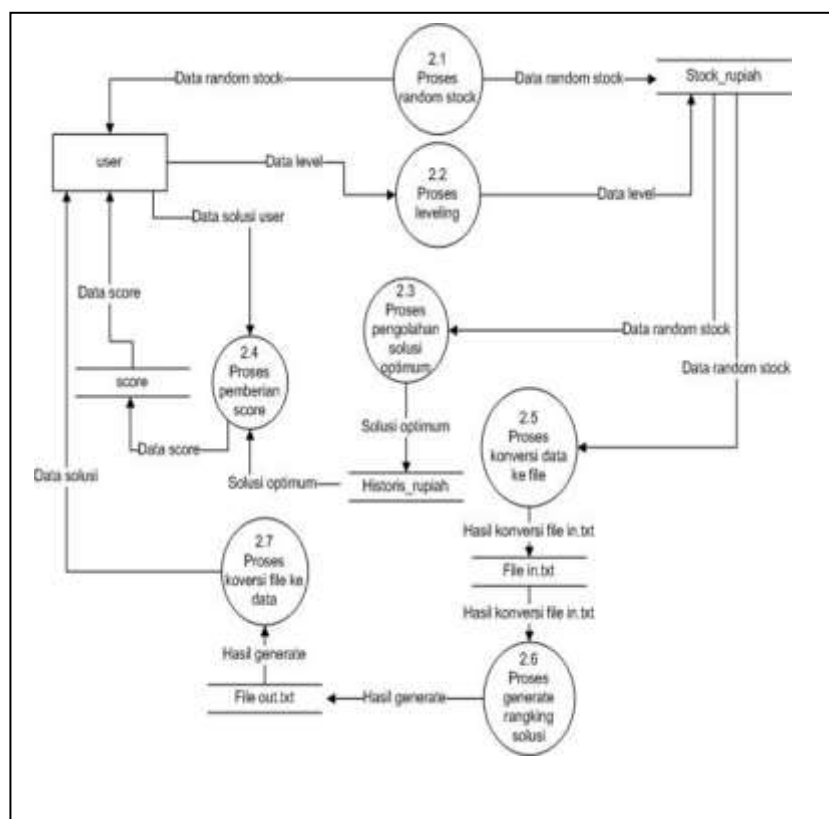
Proses, data yang sudah di input akan di proses disini yang merupakan bagian inti dari sistem, karena semua proses penting di eksekusi dibagian ini. Hasil dari proses yaitu random stock, data solusi dan data score. Dan terakhir Output, yaitu bagian eksekusi yang hasilnya nanti dapat diterima oleh user, meliputi data solusi, data score, dan data ranking antar user.



Gambar 3.2 : DFD level 2 input

3.1.1. Input Data User

Pada input data user dilakukan pengolahan inputan data dari user yang berupa data user yang dimana data user tersebut akan disimpan dalam tabel user dan juga digunakan sebagai data login ke sistem.



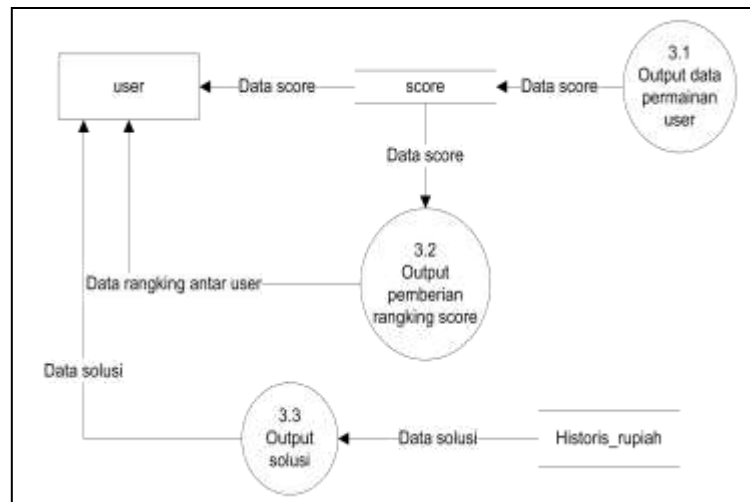
Gambar 3.3 : DFD level 2 proses

3.1.2. Proses Random Stock

Proses random stock dilakukan setelah user memilih level atau pada saat level bertambah, jadi random stock bergantung pada level atau tingkat kesulitan yang dijalankan oleh user.

3.1.3. Proses Leveling

Proses leveling merupakan penentu dari random, dimana level yang dipilih oleh user akan di jadikan patokan dalam proses random.



Gambar 3.4 : DFD level 2 output

3.1.4. Output data permainan user

Output data permainan user merupakan hasil atau nilai yang diperoleh oleh user setelah memainkan permainan berupa score.

3.1.5. Output pemberian ranking score

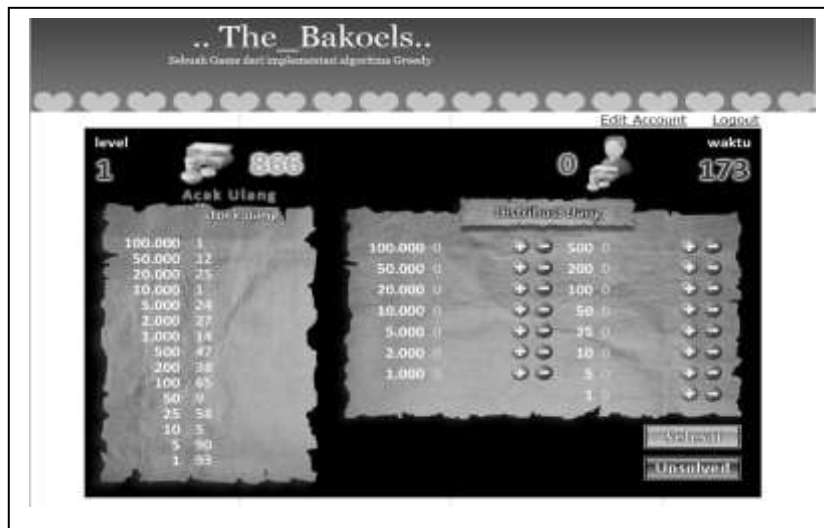
Pada pemberian ranking score antar user diambil data score semua pemain dan di urutkan dari yang tertinggi ke terendah, tetapi yang ditampilkan hanya maksimal 10 ranking score terbaik saja

3.1.6. Output Solusi

Output solusi yang bisa dilihat oleh user ada dua yaitu hasil dari proses pengolahan solusi optimum, yang juga tesimpan dalam tabel. Dan hasil dari proses generate ranking solusi yang dieksekusi oleh program C yang di tampilkan oleh aplikasi game.

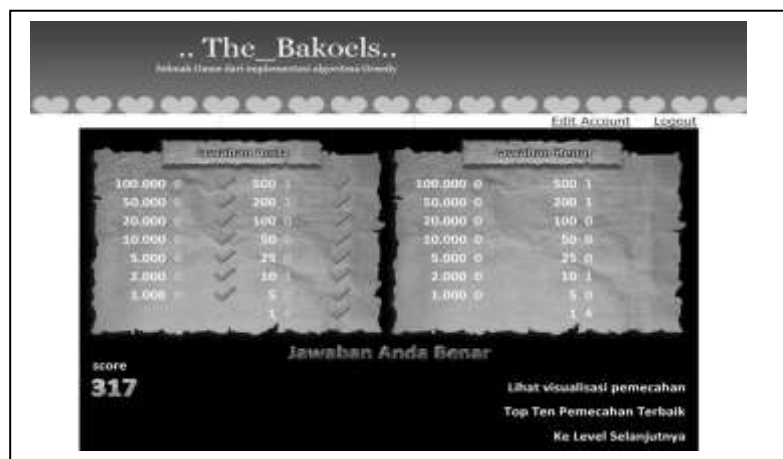
4. Implementasi

Gambar 4.1 adalah interface tampilan halaman permainan untuk user bermain. Di halaman ini user diberi soal stock uang yang di random untuk dikerjakan oleh user. Halaman ini dapat diakses setelah user memilih level dan menekan tombol next.



Gambar 4.1 : Halaman Permainan

Tampilan halaman pada saat user selesai memainkan permainan. Di halaman ini jawaban user di cocokkan dengan jawaban *Greedy*. Halaman ini dapat diakses setelah user menyelesaikan permainan atau menekan tombol selesai.



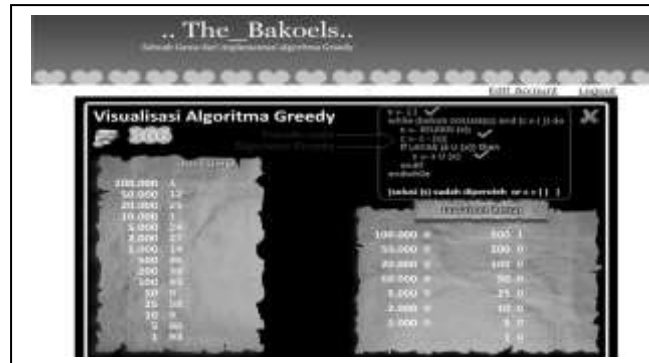
Gambar 4.2 : Halaman Selesai Permainan

Gambar 4.3 adalah interface halaman top ten pemecahan terbaik yang terdapat pada halaman selesai permainan. Jika menu top ten pemecahan terbaik di klik, akan menampilkan 10 ranking solusi terbaik yang telah dikerjakan oleh program C.

Top 10 Jawaban Benar													
100.000	50.000	20.000	10.000	5.000	2.000	1.000	500	200	100	50	25	10	5
0	0	0	0	0	0	0	1	1	0	0	0	1	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	2	0	0	1	0
0	0	0	0	0	0	0	0	3	1	0	0	1	0
0	0	0	0	0	0	0	1	0	2	0	0	0	0
0	0	0	0	0	0	0	0	3	1	0	0	0	0
0	0	0	0	0	0	0	0	3	1	0	0	0	0
0	0	0	0	0	0	0	0	3	0	2	0	1	0
0	0	0	0	0	0	0	0	2	3	0	0	1	0
0	0	0	0	0	0	0	1	1	0	0	0	0	1

Gambar 4.3 : Halaman Top Ten Pemecahan Terbaik

Gambar 4.4 adalah interface halaman visualisasi algoritma greedy dimana pada halaman ini user dapat melihat visualisasi bagaimana cara kerja algoritma *greedy* dalam mendapatkan solusi optimum untuk menyelesaikan soal pemilihan uang kembalian. Halaman ini dapat diakses dengan user menekan tombol visualisasi, tombol ini muncul setelah user mendapatkan score.

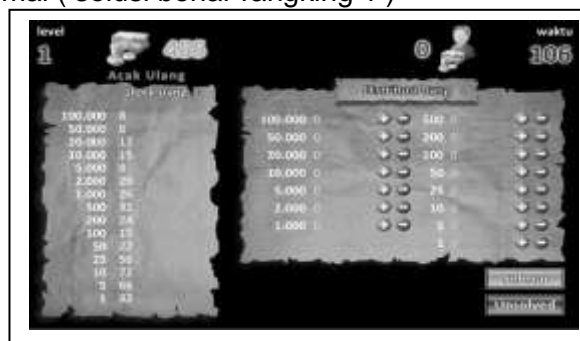


Gambar 4.4 : Halaman Visualisasi Algoritma Greedy

4.1.1. Testing (Pengujian)

a. Rancangan Pengujian

1. Solusi Optimal (solusi benar ranking 1)



Gambar 4.5 : Soal Random 1

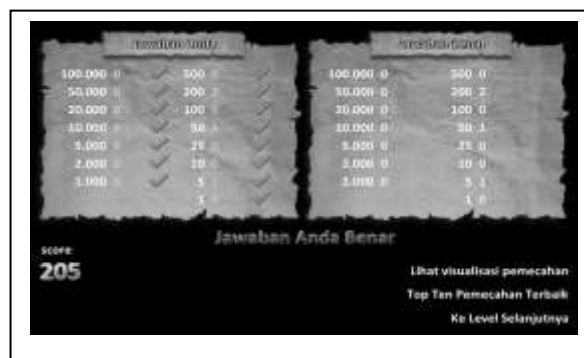
Untuk memperoleh jawaban benar dan optimal secara perhitungan adalah :

Diket : soal random 1 = 455

Tentukan : solusi optimal ?

Jawab : $200 \times 2 = 400$
 $50 \times 1 = 50$
 $1 \times 5 = 5$
 $= 455$ (jawaban benar)

Pembuktian Program :

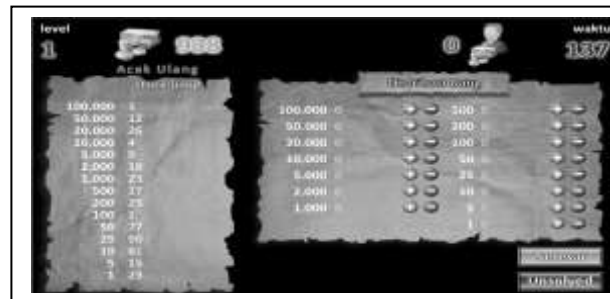


Gambar 4.6 : Jawaban Optimal

Kesimpulan : **Terbukti program berjalan dengan benar.**

2. Solusi Kurang Optimal (solusi benar rangking 2 – 10)

Misal contoh soal random :



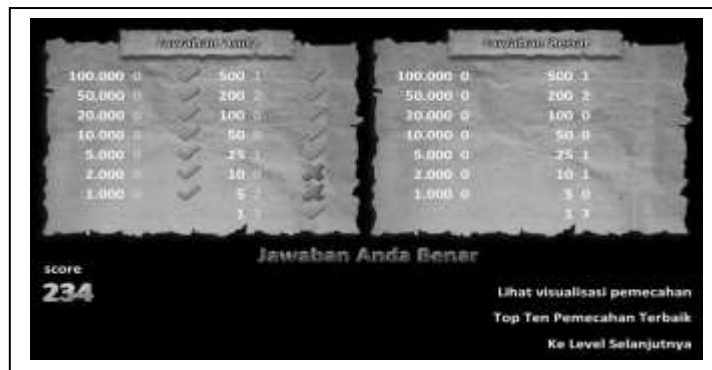
Gambar 4.15 : Soal Random 2

Untuk memperoleh jawaban benar tetapi kurang optimal secara perhitungan adalah :

Diket : soal random : 938, tentukan : solusi optimal ?

$$\begin{aligned}
 \text{Misal : } 500 \times 1 &= 500 \\
 200 \times 2 &= 400 \\
 25 \times 1 &= 25 \\
 5 \times 2 &= 10 \\
 1 \times 3 &= 3 \\
 &= 938 \text{ (benar, tetapi tidak optimal)}
 \end{aligned}$$

Pembuktian Program :

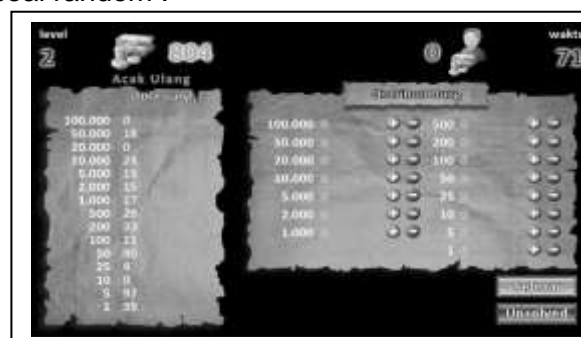


Gambar 4.16 : Jawaban Kurang optimum

Kesimpulan: **Terbukti program berjalan dengan benar, walaupun jawaban kurang optimal.**

3. Solusi Tidak Optimal (solusi salah)

Misal contoh soal random :



Gambar 4.17 : Soal Random 3

Untuk memperoleh jawaban salah atau tidak optimal secara perhitungan adalah :

Diket : soal random : 804

Tentukan : solusi optimal ?

Misal : $500 \times 1 = 500$

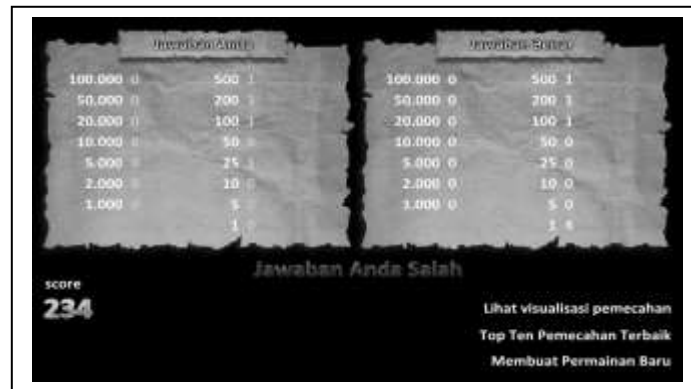
$200 \times 1 = 200$

$100 \times 1 = 100$

$25 \times 1 = 25$ +

$= 825$ (jawaban salah)

Pembuktian Program :



Gambar 4.18 : Jawaban Salah

Kesimpulan : **Terbukti program berjalan dengan benar , karena program dapat mengetahui jawaban salah.**

a. Klasifikasi Keberhasilan dalam pengujian.

- Berhasil

Pengujian dianggap berhasil, jika dalam sebuah kasus uji telah memenuhi tujuan pengujian yang telah ditetapkan, serta memiliki kinerja seperti yang diharapkan.

- Tidak Memenuhi

Pengujian dianggap tidak memenuhi, jika dalam sebuah kasus uji tidak memenuhi tujuan pengujian yang telah ditetapkan. Dalam hal ini kasus uji tersebut dinyatakan tidak memenuhi.

b. Analisa Hasil Pengujian

Dari pengujian yang telah dilakukan, dapat diperoleh kesimpulan bahwa fungsionalitas perangkat lunak dapat berjalan dengan baik. Semua prosedur untuk dapat dimanfaatkan sesuai kebutuhan pengguna.

5. Kesimpulan dan saran

Berdasarkan hasil perancangan dari implementasi algoritma *Greedy* : Suatu visualisasi dalam bentuk game edukasi berbasis web, dapat diambil kesimpulan sebagai berikut, bahwa hasil rancangan game edukasi yang merupakan implementasi algoritma *Greedy* dapat berjalan dengan baik yang dapat divisualisasikan dan di smulasikan dalam bentuk permainan berbasis aplikasi web. Saran yang perlu di berikan tampilan visual dari hasil pencarian solusi optimum menggunakan Algoritma *Greedy* oleh program C lebih dibuat menarik, tidak hanya dalam bentuk tabel saja. Variasi pemilihan mata uang asing perlu di tambahkan supaya lebih menarik dan variatif

DAFTAR PUSTAKA

1. Beatly, Gilles Brassard dan Paul. *Fundamental Of Algorithmic*. New Jersey : Prentice Hall Engelwood Cliffs, 1996.
2. Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. s.l. : McGraw-Hill, 1990.
3. Alfred V Aho, John E. Hopcroft and Jeffery D Ullman. *The Design And Analysis of Computer Algorithm*. s.l. : Addison Wesley, 1974.
4. Luqi dan Ketabchi. *A Computer-Aided Prototyping System*. s.l. : IEEE Software 5, 1988.
5. MF, Smith. *Software Prototyping: Adoption, Practice and Management*. London : McGraw-Hill, 1991.
6. Stanwick, Susan Fowler and Victor. *Application Design: Best Practices for Web-Based Software*. San Francisco : Elsevier Publisher, 2004.
7. Andriole, Stephen J. *Information System Design Principles for the 90s: Getting it Right*. Virginia : AFCEA International Press, 1990.
8. Alfred V. Aho, John E. Hopcroft dan Jeffrey D. Ullman. *Data Structures and Algorithms*. New Jersey : Murray Hill, 1999.
9. Leiss, Ernst L. *A Programmer's Companion to Algorithm Analysis*. s.l. : Taylor & Francis Group, 2006.
10. Siang, Jong Jek. *Matematika Diskrit*. Yogyakarta : Andi Offset, 2002.
11. Alsuewalyel, M.H. *Algorithms Design Techniques And Analysis*. s.l. : Publishing House Of Electronic Industry, 2002.
12. Knuth, Donad E. *The Art of Computer Programming, Vol 1,2,3*. Addison Wesley.
13. Battiti, Roberto dan Brunato, Mauro. *Reactive Business Intelligence. From Data to Models to Insight*. Trento, Italy : s.n., 2011. ISBN 978-88-905795-0-9.
14. Avedon, Elliot dan Sutton-Smith, Brian. *The Study of Games*. s.l. : John Wiley, 1971. ISBN 0471038393.
15. Salen, Katie dan Zimmerman, Eric. *Rules of Play:Game Design Fundamentals*. MIT : MIT Press, 2003. ISBN 0-262-24045-9.

Biografi Penulis

Wijanarto, Menyelesaikan pendidikan Magister Ilmu Komputer (M.Kom) tahun 2006 di Universitas Gajah Mada, Yogyakarta. Staf pengajar di Fakultas Ilmu Komputer Universitas Dian Nuswantoro Semarang

Eko Safitri UH, Menyelesaikan pendidikan Sarjana Ilmu Komputer (S.Kom) tahun 2011 di Fakultas Ilmu Komputer Universitas Dian Nuswantoro Semarang